School of Computing & Information Technology

CSCI262/CSCI862 System Security
Spring 2021

Assignment 3 (12 marks, worth 12%)

Due 24 October, 2021 23:55

Intrusion detection system

You are to implement, in Python or C/C++ or Java, an intrusion detection system in the sense of that described in the lecture notes. We are assuming all activities are associated with the same user. No GUI implementation is required in this assignment. **Your code must work on Capa. Code cannot be compiled or executed may receive a zero mark.**[1]

You must provide compilation instructions for your program and the produced program should be named `IDSE`. It should run with the command

```
IDSE Events.txt Base-Data.txt Test-Events.txt
```

where the three files do not need to have those names but will follow the formats given below. A Java program should run with Java in front of the command.

There are some files that you will work from. An example of each and the generic structure of each are provided. Examples of the required output will be demonstrated.

A specific example of the first file, `Events.txt` is

```
5
Logins:2:Total time online:1:Emails sent:1:Orders processed:1:
Pizza's ordered online:0.5:
```

The general format is

```
Number of monitored events
Event-1:Weight-1:Event-2:Weight-2:Event-3:Weight-3:Event-4:Weight-4:
Event-5:Weight-5: ....:
....
```

[1] In case you cannot connect to Capa, please contact me as soon as possible.

Only four events are recorded per line. There will be multiple lines, as many as are necessary to give the details of the specified `Number of Monitored Events`. `Number of Monitored Events` will be a positive integer no greater than 20.

The second file, `Base-Data.txt` contains data based on measuring output associated with the events described in the file `Events.txt`. Part of a specific example of the second file, associated with the specific example of the first file above, is:

```
3:290:61:148:2:
2:370:50:173:4:
5:346:87:131:1:
.
.
.
3:325:60:145:5:
```

The general format for a single line of the file is

```
Measure-Event-1:Measure-Event-2:Measure-Event-3:....:Measure-Event-Number of monitored events:
```

Each line contains the measures from a particular day. Each entry is the value associated with that event on a particular day.

The third file, `Test-Events.txt`, has the same form as `Base-Data.txt`, but each line is to be processed and tested against the base profile. Each corresponds to a days activity. These lines are not to be taken into account in determining the baseline behaviour of the user. A specific example is:

```
5:387:75:120:2:
1:123:25:50:5:
```

The general format for a single line of the file is

```
Measure-Event-1:Measure-Event-2:Measure-Event-3:....:Measure-Event-Number of monitored events:
```

What do you need to do?

1. Read in the first two files, produce a base profile, and report it, as in the example below. As mentioned earlier, this is all assumed to be for a single user. You have been given the event names and the weights in the first file, `Events.txt`. You need to calculate the average and standard deviation (stdev) based on the data given in the second file, `Base-Data.txt`. The average and standard deviation should be listed to two decimal places only.

Event	Average	Stdev	Weight
Logins	4.50	1.25	2
Total time online	287.15	42.12	1
Emails sent	65.40	30.71	1
Orders processed	150.73	20.13	1
Pizza's ordered online	2.03	1.06	0.5

   Your output doesn't need to follow this exact format but it should be clear.

2. Calculate a threshold for detecting an intrusion. The threshold is 2∗(Sums of weights). This should be reported. For the table above we have

   ```
   Threshold                    11
   ```

   Your output doesn't need to follow this exact format but it should be clear.

3. For each item in the third file, `Test-Events.txt`, you need to report on whether there is an intrusion detected. You do this by measuring adding up the weighted number of standard deviations each specific tested event value is from the average for that event, where the standard deviation and average are those you have generated from the base data and reported. For example, if 2 Logins occur in a day, we are 2 standard deviations from the average. Since Logins have a weight of 2 this contributes a distance 4 to our measure.

   For each event you should report the distance value and whether or not an alarm is raised.

   ```
   Line 1 -- 5:387:75:120:2:          Distance: ...          Alarm: No
   Line 2 -- 1:123:25:50:5:           Distance: ...          Alarm: Yes
   ```

   Again, your output doesn't need to follow this exact format but it should be clear.

Notes on submission

1. Submission is via Moodle.
2. Include the compilation instructions with your submission (i.e., provide a readme.txt file).
3. Late submissions will be marked with a 25% deduction for each day, including days over the weekend.
4. Submissions more than three days late will not be marked, unless an extension has been granted.
5. If you need an extension apply through SOLS, if possible **before** the assignment deadline.
6. Plagiarism is treated seriously. Students involved will likely receive zero.